Chapter 2: Background

This chapter presents background on the agent paradigm and necessary definitions of what an agent and a Multi-Agent System (MAS) are. The chapter also presents an overview of the various origins of the agent paradigm. Current research is then described and a comparison is made between current approaches. An introduction and rationale for an agent system is given in section 2.1. Section 2.2 provides necessary definitions of an agent system as well as various classification methods for an agent. Section 2.3 extends agent systems into multi-agent systems and proposes a MAS classification method. Section 2.4 discusses problems related to MASs. Origins of agents and MASs, together with an overview of current research are given in section 2.5. Section 2.6 concludes this chapter with a summary.

2.1 Introduction

The research field of cooperating, embedded, heterogeneous multi-agent systems is becoming more mainstream than ever before. Many new MAS applications are simulated [173] and built [36]. This is hardly surprising, considering that the evolution of the paradigm for the development of computer systems has always lead to more independent, loosely-coupled modules. Initially, software development has relied on machine dependent, low-abstraction level, assembler programming. Procedural programming, as exemplified by 3rd generation programming languages (e.g. Pascal, C, etc.), was a major improvement on assembler-type programming languages. The onset of the object-oriented programming paradigm (e.g. C++, Java, etc.) heralded another qualitative shift towards more independent, reusable components. Today, mainstream information technology has fully embraced even more independent modules that interact through mechanisms such as CORBA, DCOM etc. Many researchers view agents as an extension of Component Based Software Engineering CBSE [151][81]. The evolution of CBSE can be illustrated as in figure 2.1.

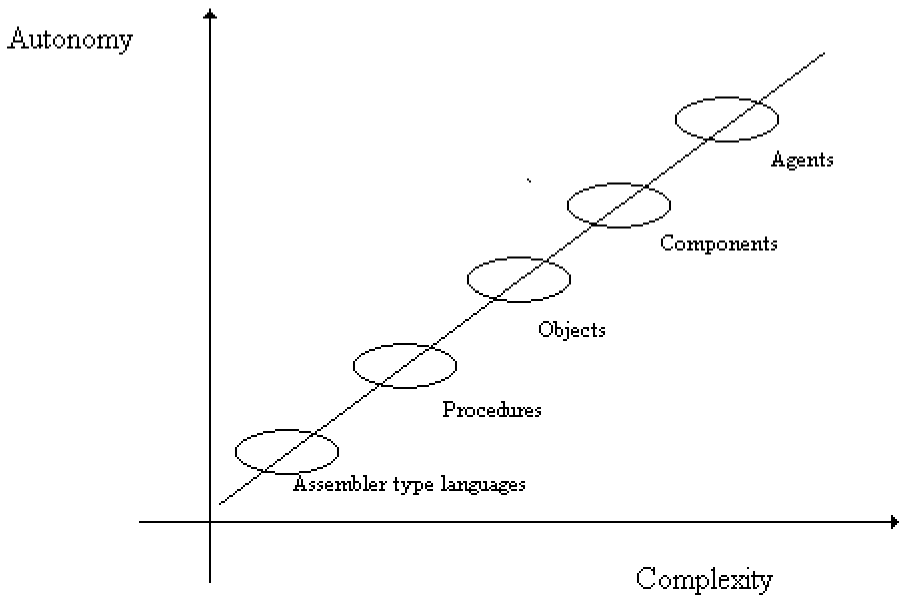

Figure 1. Evolution of Component Based Software Engineering

The current paradigm shift is towards independently interacting components that will have the property of self-organisation in order to solve a problem that is defined in general terms only. Those components are agents. Indeed, the probability is that most new IT development and products will, in one form or another, contain embedded agents [83]. The generalisation of agent systems resulted in the appearance of Multi-Agent Systems (MAS). MASs offer all the advantages of parallel distributed systems. The parallelism of MASs allowed application of the agent paradigm to an even greater set of problems that exceeded the capabilities of a single agent.

As the complexity of problems to be solved grew, so did the complexity of the paradigms that offer a solution to these complex problems. By introducing parallel distributed systems, the need for coordination between agents became obvious. The fact is that more and more complex Artificial Intelligence (AI) techniques are applied in the implementation of agents and MASs. AI techniques, as a general rule, are heuristics. It would be close to impossible to try to test a MAS using a white box approach. In other words, traditional tests and models are becoming rapidly obsolete as more AI based MASs are being developed and deployed. However, there is a need for some kind of model that will allow the computer scientist to design and test more complex systems. This has resulted in the recent, radically different approach to the development of such models. As the systems are growing more complex, (nearly) approaching the complexity and diversity of simpler biological systems, a possible solution to the problem of moving towards the next paradigm is an AI-based MAS model using biological, social and organisational models. The idea is not exactly new [70][117], but recently it received momentum from the fact that some well-known researchers are proposing new MAS models based on social and behavioural models [141][53].

2.2 Agents: Definitions and Classifications

2.2.1 Introduction

Agent systems are rapidly becoming mainstream in the IT industry. The introduction to this chapter (see section 2.1) presented a reasoning for agent systems that mainly considered software engineering issues such as complexity hiding and the efficiency of parallel distributed systems. That is not the only reason for the increasing popularity of agent systems. The increase in complexity of tasks that are performed is not only imposed on the software systems developer. The complexity of tasks is affecting the end-users to an even greater degree, due to the fact that the user often has to perform a complicated set-up, and use complex operations in order to solve a problem. Considering the fact that computers are no longer viewed as tools for specialists only, the drive is to make efficient use of computers, even by inexperienced users.

One way of achieving this is to have intelligent helpers (agents) that help users to achieve a desired outcome. It is debatable if these intelligent helpers are agents in the true sense of agency (as proposed in section 2.2.3), as they have limits imposed on their autonomy, and collaboration between intelligent helpers and other agents systems (excluding users) is often limited. Nevertheless, proliferation of such agents is rapid. Some examples of such agent systems are Microsoft Office Assistant, Information Filtering Systems [200], intelligent web search engines [190][192] etc.

2.2.2 Agent Definitions

As is very common in the field of Artificial Intelligence, there is no standard definition of an agent. Instead, it seems that almost every major research and survey yields yet another definition. For the sake of completeness, some of the definitions are presented below.

An agent is:

"a computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives" [197].

Others define an agent as:

- "a system that independently handles parts of the problem based on small independent knowledge bases" [82].

- "an autonomous entity that interacts with the environment, and adapts its state and behaviour based on interaction" [139].

- "an agent is a computational entity which:
  - acts on behalf of other entities in an autonomous fashion
  - performs its actions with some level of proactivity and/or reactiveness, and
  - exhibits some level of the key attributes of learning, cooperation and mobility" [80].

This thesis does not propose a new definition of an agent. Instead, an effort is made to extract the common characteristics of an agent from various agent definitions, as given in the next section.

2.2.3 Characteristics of Agents

As noted in the previous section, there is no common definition of an agent. However, it seems that most researchers agree on certain characteristics of agency. For the purpose of this thesis, a computational entity is considered an agent if it possesses the following characteristics:

- **Autonomy**: An agent has its own beliefs, plans and intentions and it can accept or refuse a request.
- **Interaction**: An agent interacts with its environment. The agent can change the environment via its actions and the environment can change the agent's actions.
- **Collaboration**: An agent must be able to collaborate with other agents in order to achieve a common goal.
- **Learning**: An agent must have the ability to learn, based on previous experience from its interaction with the environment.

It is important to note that some of the quintessential agents and agent architectures do not fully have all of the proposed characteristics. Most notably, agents in the subsumption architecture [31] do not have full collaboration and learning characteristics, while agents in behaviour based architectures [113] do not "consciously" collaborate. The proposed set of characteristics can be seen as the result of evolution of the desired characteristics for an agent and represents the current mainstream approach to agency.

The prospects of having a standard definition of an agent are as good as having a standard definition of an intelligent system.

The next section presents some of the ideas that have contributed to the creation of an agent-oriented systems paradigm.

2.2.4 Agent Classification Schemes

Classification schemes for agents are relatively unexplored. Some classification schemes are implicitly given in various agent surveys [88][80] and some explicitly [138]. This thesis presents classifications based on agent reasoning model, agent key attributes [138] and agent paradigm origin. The following sections discuss each of these classification models.

2.2.4.1 Reasoning Model Classification

Classification based on an agent's reasoning method is not new. Despite the fact that classification based on reasoning method is not new, there is still no consensus on the exact naming of the two main paradigms that form the basis of this classification. The two main paradigms that form reasoning method classification are symbolic and sub-symbolic paradigms. Symbolic and sub-symbolic paradigms are respectively referred to as traditional and connectionist, or deliberative and reactive paradigms. These are all different names for the fundamental division between two different approaches in the field of AI.

According to reasoning method, agents can be classified into the following three distinctive groups:

- **Symbolic Reasoning Agents,** which utilise a traditional AI approach based on logic calculus. Traditional AI approaches are exemplified in the majority of expert systems. The main characteristic of a symbolic reasoning agent is that it relies on symbolic representation of the real-world. Symbolic reasoning agents usually have the following components [88]:

    - A symbolic model of the world, usually represented in some form of rules such as first-order predicate logic.
    - A symbolic specification of the agent's actions, usually represented as a rule with a condition for its triggering, which consists of an antecedent (a conjunction of Boolean conditions) and a consequent (or action).
    - A reasoning algorithm that plans the agent's future actions. All reasoning-related computations usually rely on inference rules, expressed in first-order predicate calculus.

  A detailed description and critique of symbolic reasoning agents is presented in chapter 3 (section 3.2), together with some examples of such systems.

- **Sub-symbolic Reasoning Agents,** which do not maintain a world model, or if they do, a non-symbolic representation is used for a world model. Sub-symbolic agents are sometimes called reactive agents. The main objective of sub-symbolic reasoning agents is to minimise the amount of predetermined behaviour, and to create agents that exhibit intelligent behaviour based on the agent's interaction with its environment. In other words, intelligent behaviour should emerge.

  The main characteristics of such agents are that they do not maintain a symbolic model of the world and usually do not communicate with other agents. The consequences are that a sub-symbolic agent's reasoning is based on interaction with the local environment.

  Despite the well-documented shortcomings of sub-symbolic agents [84][95], some of the sub-symbolic agent implementations have achieved spectacular results, albeit in very specific domains [30]. A more detailed description of this architecture and its critique is presented in section 3.3.

- **Hybrid Reasoning Agents,** which combine the characteristics of symbolic and sub-symbolic agents. Shortcomings of both symbolic and sub-symbolic models have become apparent fairly early and they are discussed in greater details in chapter 3. Various hybrid models were proposed that try to exploit the best of both approaches, such as MACTA [11][10], InteRRaP [130][129] and Touring Machines [63]. Most hybrid architectures are layered architectures, where lower layers are simpler (reactive or behavioural) and upper layers are more complex, providing symbolic reasoning capabilities, as well as mechanisms for cooperation between various agents.

This thesis assumes agent classification based on reasoning model.

2.2.4.2 Agent Key Attribute Classification

Nwana presents a typology based on the premise that agents can be classified along several ideal, primary attributes that the agent should exhibit [138]. The minimal set of identified attributes includes autonomy, learning and cooperation. If compared with the desired characteristics of an agent, as presented in section 2.2.3, it is indicative that the characteristic of interaction with the environment is missing. The classification according to agent key attributes divides agents into seven distinctive groups:

- Collaborative (Cooperative) agents that are interested in cooperation with other agents. According to the agent's characteristics adopted in this thesis, all agents should be collaborative.
- Interface agents are agents developed to facilitate user-machine interaction.
- Mobile agents are agents capable of moving through physical environments, for example, robots.
- Information/Internet agents are agents mainly used for retrieval and search of information on the Internet.
- Reactive agents are agents that do not maintain any internal environment representation, and simply react on stimuli received from the environment.
- Hybrid agents that combine reactive agents with deliberative thinking.
- Smart agents were not clearly defined by Nwana but implicitly they should be "super-agents" combining collaboration, deliberative thinking and learning capabilities.

The shortcomings of the proposed classification are numerous but the classification is overviewed here for the purpose of completeness. The presented classification is a combination of divisions according to the agent's tasks and the agent's architecture and as such may lead to confusion as an agent can belong to more than one category (classified in one instance as what it does and in another instance as how it does it) according to this classification scheme.

Furthermore, some major categories are missing. For example, if the classification includes reactive and hybrid agents, it should surely include the symbolic reasoning (or deliberative) agents. Another such category is that of self-interested agents, as not all agents are collaborative. Due to the above-mentioned shortcomings agent key attribute classification is of limited value for the purpose of this thesis, and it is not used in this thesis.

2.2.4.3 Paradigm Origin Classification

There were many contributing origins to the field of agent systems and MASs. Various overviews [88][138][80] have investigated the origins of agent paradigms. This section overviews a classification scheme based on a combination of these overviews.

Agents can be classified according to their original paradigm background into

- *Artificial Intelligence (AI) agents*, which is the main contributor to the field of agent systems [88]. Various sub-fields of AI have been incorporated into agent systems, such as artificial life, swarm intelligence, distributed artificial intelligence, traditional AI approaches and evolutionary computation. The AI contribution to current agent research is largely due to the scientific research done at academic institutions and various agents have their origins in AI research.

- *Object Oriented Programming (OOP) agents* – Many agent architectures are developed using the OOP paradigm [88][80]. It is not surprising that OOP is an origin of agent paradigm, considering that agents are the natural evolution of CBSE, as discussed in section 2.1. Frequently, objects are used as a starting point for an agent implementation because both agents and objects have shared characteristics, such as encapsulation and data hiding.

- *Machine-Man Interface agents* – Machine-man interface research is receiving strong impetus, based on industry and consumer demand. This is due to the continuous increase in complexity of tasks that today's and future users will face. Complex tasks need to be automated and streamlined. Agents are often used to help and guide users by being adaptive [80].

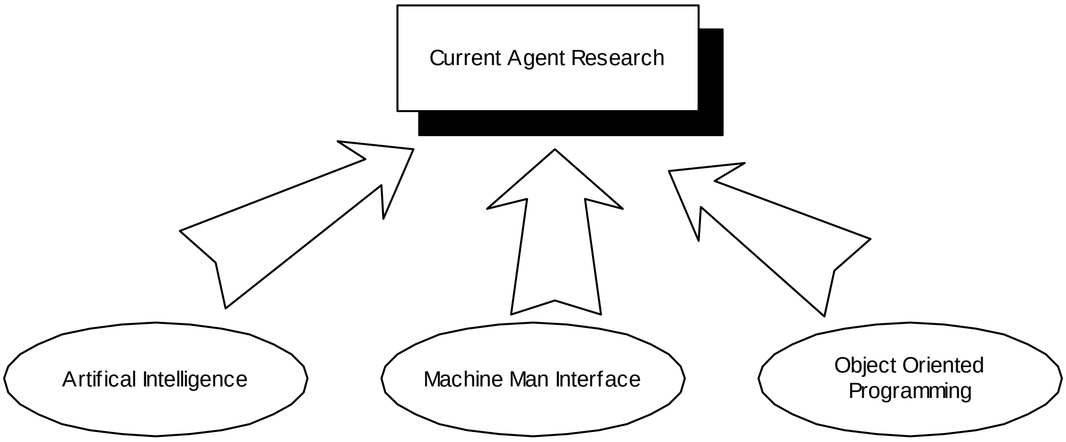

Figure 2. Origins of Agent Paradigm

- *Robotics* – Since the early age of civilisation, mankind was obsessed with creating tangible, real-world, intelligent, autonomous artefacts. Various explanations for such obsession can be given, but certainly some early inspirations, as described in literature, can be found in religion (e.g. the creation of golems, powered by the word of God [188]). Other reasons were economical (e.g. creation of the intelligent labourers as described in Karl Chapek's novel "Rossum's Universal Robots" [39]) and scientific (as in Mary Shelly's "Frankenstein" [169]). Today, robotics, the science of creating such artefacts, has come a long way from their literary and religious origins. Robotics, as a scientific discipline, often assumes a holistic approach to agent technology. It combines some of the disciplines above, such as software engineering, AI, artificial life, electronics, mechanics and other not so obviously related disciplines, such as organisational science, sociology, biology, etc. The product of robotics is a robot – the ultimate agent.

The origins of agent paradigm are illustrated in figure 2.

2.3 Multi Agent Systems: Definitions and Classification

2.3.1 Introduction

A system that consists of multiple agents is called a Multi-Agent System (MAS). A MAS is a generalisation of an agent system where the main advantages of agents can be further exploited, namely an agent's ability to execute both autonomously and in parallel. A MAS is ideally suited for problems that can be either executed in parallel or that can employ multiple problem-solving methods. However, the advantage of a MAS approach to problem-solving and parallelism does come at a price: interaction problems between autonomous agents exists, including cooperation (working towards a common goal), negotiations (coming to an agreement) and coordination (avoiding harmful interactions between agents). Some definitions of MASs taken from literature are given in the next section.

2.3.2 MAS Definitions

There are various definitions of a MAS. For the purpose of this thesis only a few are presented. A MAS can be defined as a loosely-coupled network of problem-solvers that work together to solve problems that are beyond the individual capabilities or knowledge of each problem-solver [25].

Other authors keep the definition much simpler: a MAS can also be seen as a society of agents [204][80].

Wooldridge and Jennings propose a rather strict definition of a MAS that is based on MAS characteristics [88]:

- Each agent has incomplete information or capabilities for solving the problem, thus each agent has a limited viewpoint.
- There is no global system control.
- Data is decentralized.
- Computation is asynchronous.

2.3.3 Characteristics of MAS

This thesis proposes a slightly relaxed definition of MAS characteristics based on [88]:

- Each agent in a MAS can have complete or incomplete information about the problem or capabilities to solve the problem.
- There is no global rigid control system. However, there can be a global coordinating system, such as a supervisor.
- A complete set of data can be partially or fully decentralised.
- Computations are executed in parallel.

2.3.4 MAS Classification Schemes

Various classification schemes of MASs are in existence [88][138][80][160]. In this section, a subset of the existing MAS classification schemes is presented. Some of the presented classification schemes are generalised versions of agent classification schemes, while others are based on properties applicable only to MASs such as communication models.

2.3.4.1 Reasoning Model Classification

As is the case with agent architectures, MASs can be classified according to the reasoning module employed by the MAS. Using such a classification scheme, MASs can be divided into three classes: symbolic MAS, subsumption MAS and hybrid MAS. These classes are presented in chronological order of appearance.

Symbolic MAS

Symbolic architectures were the earliest to emerge as MAS [138]. This is hardly surprising if it is taken into consideration that a significant contribution to the agent paradigm came from AI planning research, which was a very active research area during the 1970s and 1980s. Symbolic MASs are based on premises of the "physical-symbol system hypothesis" [133]. Newell and Simon, defined a physical symbol system as a

"…physically realisable set of physical entities (symbols) that can be combined to form structures and which is capable of running processes that operate on those symbols according to symbolically coded sets of instructions" [133] .

The physical-symbol hypothesis then stipulates that a physical symbol system is capable of general intelligent action.

Wooldridge and Jennings define a symbolic architecture as

"[an] architecture that contains an explicitly represented, symbolic model of the world, and in which decisions are made via logical (or at least pseudo-logical) reasoning, based on pattern matching and symbolic manipulations" [197].

Symbolic architectures, as any other architecture, have their advantages and disadvantages, which are discussed in the section 3.2.2.

Typically, a symbolic MAS is based on a problem-solving method, such as STRIPS [65] that employs a symbolic, centralised model of the world. A symbolic MAS is based on the cognitive science sense-think-act cycle. The sense-think-act cycle assumes that an agent senses a change in the environment, deliberates about the change in the environment and decides on an optimal or nearly optimal course of action and, lastly, executes an action which may have an effect on its environment. In theory this sounds very good but there were problems when implemented in real-world environments. In practice, problems such as slowness of deliberation and accurate real-world modelling were experienced. Systems such as STRIPS [65] and General Problem Solver (GPS) [134] have performed extremely well in virtual worlds, where the model of the world was static and accurately given.

A more detailed discussion on the advantages and disadvantages of symbolic MAS is presented in section 4.3.

Sub-symbolic MAS

Once the limitations of symbolic MASs became obvious and theoretically proven [88], more criticism followed and the most influential critique came from Brooks [28][35]. As a complete opposite to the symbolic approach, an approach where knowledge is subsumed was proposed in the seminal work by Brooks [31]. In this purely reactive approach, knowledge is subsumed in condition-action pairs. Intelligence is treated as a "side-effect" of an agent's interaction with its environment.

The subsumption architecture employs no symbolic knowledge at all, hence there is no model of the world. It assumes that intelligent behaviour emerges from interaction between more primitive behaviours represented, essentially, as action-reaction pairs.

The subsumption architecture has been surprisingly successful, despite its apparent simplicity, but there are serious disadvantages of this architecture. An obvious problem is that, because of the lack of a world model, every agent decides on its actions based on information from its local environment. Therefore, there is no coordination as such, actions are only locally optimal, and overall behaviour is not easily understood.

An additional problem is that there is no effective way for an agent to learn from experience, as there is no direct feedback loop from consequences to actions. Details on this architecture are presented in section 4.4.

By the 1990s it was accepted that the subsumption architecture may be applicable to certain problem domains, such as modelling of insect behaviour [5], but it was not suited as a general architecture. An attempt to reconcile symbolic and sub-symbolic approaches resulted in the next class of MAS, i.e. the hybrid MAS.

Hybrid MAS

Hybrid MAS is a result of trying to use the best of both worlds, i.e. symbolic and subsumption MAS. Two main problems of the symbolic architecture, namely its slowness and the problem of accurate world modelling were related to its interaction with its environment. Most of the strengths of the symbolic approach come from its deliberative and planned approach to acting on stimuli from the environment. On the other hand, the main strength of the sub-symbolic architecture stems from its efficient interaction with its environment and the main weakness is the fact there is no efficient, goal-driven interaction between agents. A typical hybrid system uses both symbolic and subsumed knowledge and exploits the strengths of each approach.

Typically, a hybrid MAS is a layered system, where different layers use different knowledge representations. The higher levels are based on symbolic knowledge reasoning, while lower levels are usually implemented using a sub-symbolic approach. A layered MAS exhibits symbolic planning and coordination, coupled with fast, efficient interaction with the environment. An example of a hybrid MAS is Multiple Automata for Complex Task Achievement (MACTA) [11][10] that utilises a symbolic planner as a symbolic component, while a sub-symbolic component is implemented using the Behavioural Synthesis Architecture (BSA) [106]. MACTA is presented in greater detail in section 4.5.

2.3.4.2 Cooperation Level Classification

With the appearance of MASs, the issue of avoiding negative interaction (or conflict) by means of negotiation and the issue of cooperation by means of coordination became very important. The potential for exhibiting negative interaction is due to the autonomy of agents, who may have their own beliefs, desires and intentions that are not necessarily shared between all agents. If agents' intentions are conflicting, a conflict may arise between the agents in a MAS. Classification of MASs based on the level of cooperation was proposed in the late '80s [24]. The cooperation based classification scheme has been adopted by other researchers [160][36]. According to level of cooperation between agents, MASs can be divided into:

Cooperative Multi-Agent Systems

Cooperative MASs, historically the first to appear, have their background in early Distributed Artificial Intelligence (DAI) [88]. In cooperative MASs, the emphasis is not in optimising the performance of an individual agent, but that of the whole system. This class of MAS roughly corresponds to a symbolic MAS, as symbolic MASs often employ symbolic representation and cooperation enabling techniques that rely on symbolic representation of the world model. The consequence is that a global world model must be maintained.

The main focus of research in cooperative MASs is that of coordination between the agents [80].

Self-Interested Multi-Agent Systems

The emphasis of self-interested MASs is on improving performance of a single agent, hoping that improvement in the performance of an individual will lead to improvement in performance of the whole system. Unfortunately, agents may be openly antagonistic or they may exhibit conflicting behaviours. The problem is further compounded if there is no means of direct communication [113] or no communication at all [28].

When it comes to interaction between agents, the main areas of interest for self-interested MASs are that of conflict resolution and negotiation, assuming, of course, the existence of a communication channel between agents.

2.4 Problems with Multi-Agent Systems

The MAS paradigm is very promising, but it has its own problems that can be broadly divided into theoretical and practical problems. Theoretical problems relate to the interaction between the agents, while practical problems are related to the scalability to real-world environments and lack of formal methods and frameworks for agent development. Theoretical problems related to the interaction between agents are the main focus of this thesis and a new approach to coordination is presented in chapter 6 of this thesis.

Although the main focus of this thesis is on a specific coordination approach, the practical problem of inadequate frameworks for development of agents (and specially robots) is also addressed through the proposed new hybrid robot architecture (chapter 5).

2.4.1 Interaction Between Agents in MAS

Agents in a MAS perform their tasks in a shared environment. The agents not only interact with the environment, but with other agents as well. A simple example would be a robot scout that can detect an obstacle that can, for example, be either a wall (environment) or another robot (agent). The interaction between the agents in a MAS can be positive (resulting in cooperation) or negative (resulting in conflict). In order to address and facilitate cooperation, a MAS needs to have a coordinating mechanism. The problem of negative interaction between agents is very serious. To avoid and resolve conflicts, a MAS needs to have a negotiation mechanism. This section presents an overview of coordination and negotiation mechanisms.

2.4.1.1 Coordination Mechanisms

Cooperation allows agents in a MAS to solve problems that exceed an individual agent's characteristics. Coordination in a MAS is crucial to allow the exploitation of one of the main benefits of MASs, namely cooperation. Coordination is a problem not unique to computer science. The coordination problem is present in many different sciences, mainly in social sciences such as sociology, anthropology, organisational sciences etc. Coordination of biological systems such as ant colonies and swarms are studied in biological-related sciences [89], but also in certain AI fields such as swarm intelligence [61]. Coordination mechanisms can be classified according to their origin as follows:

Organisational Coordination

Probably the simplest way of coordinating agents is by establishing a relatively strict hierarchical architecture that prescribes roles and protocols for agents to communicate with others within the hierarchy. Examples of organisational coordination architectures are MAGMA [181] and to a certain degree SMAK [93]. A shortcoming of the organisational coordination approach is that it may prohibit optimal task allocation due to extreme specialisation of each agent in the system. The hierarchical approach is also somewhat contrary to the idea of autonomous agents as they are coordinated from a central system and are not truly autonomous but dependent on the central system.

Contracting as Coordination

The contracting approach is based on an agent opening an auction for task allocation and other agents bidding for the executing the task. The idea of using an auctioning mechanism in AI is not new [79], with one of the most applied coordination techniques, the Contract Net Protocol (CNP), based on auctioning [49]. Although the authors, Smith and Davis, refer to CNP role as a negotiation tool, the view adopted in this thesis and by other researchers [88] is that it is really a coordination tool. The CNP assumes that agents fulfil separate roles, the one of bidder and the other of auctioneer, which has elements of an organisational coordination approach but the roles are not predefined and an agent can assume both roles. CNP offers a simple yet powerful, mechanism for coordination. The main critique of this approach is that it assumes a market economy [52], where there is an abundance of bidders and that the task should be relatively well known.

Society-Based Coordination

In view of one of the proposed definitions of a MAS as “a society of agents” [204] and in the recent work of some prominent researchers [141][53], it makes sense to let social regulations coordinate a MAS. Social regulations can be divided into social rules that regulate an individual agent’s behaviour and social structures that regulate interaction between agents.

A more detailed discussion of coordination methods based on the theory of organisational sociology is presented in section 6.3.

2.4.1.2 Negotiation Mechanisms

The purpose of negotiation mechanisms is to prevent or resolve conflicts between the agents in MASs. There is still no consensus in the MAS research community on the importance of negotiation. While some architectures ignore negotiation completely [28], other researchers propose fairly complicated mechanisms for negotiation [151]. The proposed new approach presented in chapter 6 relies on coordination in order to prevent a problem instead on negotiation to resolve a problem. As the focus is on coordination, only an overview of negotiation techniques is presented in this thesis. Based on the classification given in 2.3.4.2, negotiation mechanisms can be divided into the following categories:

Competitive Negotiations

Competitive negotiations are particularly applicable to self-interested MASs where agents do not necessary cooperate; instead, the agents try to achieve their own goals. An example of a competitive negotiations environment is an agent trading in an e-commerce environment where negotiations (agreeing on a price) are done between self-interested, competing, autonomous agents [181][40]. There are various techniques used for competitive negotiations, such as game theory-related techniques [49], auctioning [181] and contracting [37].

Cooperative Negotiation

Cooperative negotiation mechanisms are applicable to cooperative MASs, where agents are willing to collaborate. This approach should be utilised when it is absolutely critical to avoid conflict, for example in the domain of air-traffic control [104][41]. The majority of systems that utilise cooperative negotiation are based on the Belief-Desire-Intention architecture [152], where negotiation is seen as updating and changing an agent's belief.

2.4.2 Scalability of MASs

Scalability of MASs to real-world problems can be viewed in many different ways. For example, it can be said that scaling up from a simulated stock exchange e-commerce trading system to a real-world stock exchange, a live e-commerce trading system can be a problem. After all, events in a real stock exchange environment are usually unpredictable.

Specifically, in the case of embedded agents (robots), scaling of systems that work very well in a simulated environment to real-world embedded agents has proven to be very difficult [34]. Initially, agents and MASs were built using a traditional symbolic approach to artificial intelligence. Although there were some success stories, such as STRIPS (Stanford Research Institute Problem Solver) which has been successfully tested on a real robot [65], it seems that symbolic reasoning cannot be the sole mechanism for the development of complex multi-agent systems [28][112]. The main reason for this is that deliberative, symbolic reasoning takes too much time in real-world environments due to the combinatorial explosion in potential decision-making processes in case of too many variables, which are usual characteristics of real-world environments.

2.4.3 Lack of Formalism

Because the agent oriented paradigm is still relatively new, research on standard design principles or standard frameworks is still limited [86][204][54]. While these approaches have been successful in standardising some of the explored domains, there is still no standard framework for the development of a MAS. The main reason why a unified framework does not exist is the variety of MASs and their application. Agents and MASs are applied in various domains ranging from Internet search agents (softbots), emulation of credible creatures in virtual reality [18] to envisaged interplanetary exploration [173]. A good example of the lack of formalism is the various means of communication mechanisms in MASs, ranging from simple 1Hz 6 byte broadcast messages [113] to the implementation of recommendations of the Knowledge Sharing Effort (KSE) group [132] that has resulted in sophisticated models such as the Knowledge Query and Manipulation Language (KQML) [66][99] and Agent Communication Language (ACL) [166].

2.5 Origins of the Agent Paradigm

The concept of an agent did not appear suddenly. As indicated in the introduction, it can be said that agents evolved from the CBSE paradigm. However, CBSE was not the only origin of the agent paradigm. In fact, the agent paradigm has evolved from four main contributing fields, namely: Artificial Intelligence (AI), Object Oriented Programming (OOP), Machine-Man Interface Research and robotics as shortly discussed in section 2.2.4.3. Each of these origins is discussed next in more detail.

2.5.1 Artificial Intelligence

The aim of AI research is to produce intelligent artefacts. Intelligence is closely linked to the ability to learn. If these artefacts are situated in an environment and can interact with the environment, it seems that the natural aim of AI research is to produce agents. Nevertheless, agents have been ignored by mainstream AI for a surprisingly long period of time. A possible explanation for this anomaly is that AI researchers were too busy improving and investigating the AI components of a system (such as machine learning, or interaction mechanisms between software and its environment), without attempting a synthesis that would deliver a true agent system. An additional contributing factor is that AI, until relatively recently (as a point of reference, consider the re-emergence of Artificial Neural Networks (ANN) in the early eighties), was almost exclusively dominated by the symbolic reasoning paradigm as embodied in expert systems.

MASs are closely related, by virtue of being a collection of distributed, intelligent agents, to the field of Distributed Artificial Intelligence (DAI). DAI has been traditionally divided into two main groups [25]:

- Distributed Problem Solving (DPS), that considered how a problem can be solved by a number of modules that cooperate in dividing and sharing knowledge.
- Traditional MASs that are usually restricted to one type of replicated agents, also known as homogenous MASs.

A turning point was reached in 1980 at the first DAI workshop at MIT where it was decided that the aim of DAI is not to optimise low level parallelism issues, such as distributing workload between numerous processors or how to improve parallelism of algorithms, but to find how intelligent problem solvers can interact in order to solve problems that cannot be solved by a single intelligent problem solver [88].

2.5.2 Object-Oriented Programming

The similarities between an object and an agent are so obvious that it is not surprising that object-oriented programming (OOP) is one of the major origins of agent research. Both an agent and an object interact with its environment (via messages in traditional OOP), collaborate within the system (either using messages or methods) and, if considered learning in loose terms, both can learn (an object can "learn" by maintaining its internal state by means of protected and private data). These similarities are, however, misleading. There is a significant difference between agents and objects. Objects are not autonomous. Objects do not have their own goals, intentions and beliefs. In other words, the object represents an ideal body; the agent brings the reasoning. In a sense, the agent paradigm can be seen as a further evolution of OOP.

2.5.3 Man-Machine Interface

As the tasks that a user needs to perform on a computer become more complex, the way that a user interacts with a computer system becomes more time consuming and more cumbersome. Ideally, a user should just give the instructions on what he/she wants to achieve, without necessarily explaining in minute detail how to do this. For this to be achieved new ways of interacting with computer systems are necessary. The man-machine interface research field is mainly interested in new ways of interacting with computer systems. One of the ways to streamline man-machine interfaces has lead to development of computer programs that cooperate with the user and that help the user to achieve what he/she really wants without explicitly instructing a computer system what to do. These computer programs need to be, at least partially, autonomous, need to interact, learn and collaborate. Such computer programs satisfy most of the characteristics of an agent. These agents are usually referred to as adaptive user interfaces or intelligent interface agents [80].

The tasks of intelligent interface agents can be divided into three groups based on the roles that the agents perform [80]:

Information filtering agents

The amount of, often unwanted, information presented to a user is increasing daily. This phenomenon is referred to as information overload. The role of an information filtering agent is to reduce the information overload based on user preferences. Filtering rules can be based on rules that the information filtering agents learn by "observing" the user's habits [190]. Alternatively, rules can explicitly be stated by the user, although the notion of agency in the second scenario is questionable. An example of an information filtering agent is Maxims [105] that manages user's emails and the user interface as implemented on amazon.com [190].

Information Retrieval Agents

The amount of information that is available for retrieval from the Internet is tremendous. It is no wonder that agents are employed in a role that allows a user to do an intelligent search of that vast amount of available information. Furthermore, agent controlled search can be executed in the background, collecting information from various sources and presenting results to the user, only when compiled and organised in a user-friendly format. An example of an information retrieval agent system is the Google search engine [192].

Expert Assistants

The task of expert assistants is to improve user interface efficiency by means of easier communication between the user and computer system. Expert assistants can be personified or not. Probably the most well known expert assistant is the Microsoft Office Assistant.

Adaptive user interfaces are not the only area of man-machine interface research that has contributed to the agent paradigm. Far more exotic than user interfaces and expert assistants are artificial life agents that populate virtual worlds in virtual reality man-machine interfaces, for example the Oz project at Carnegie Mellon University [65] and virtual worlds created in the MIT Media Lab [21].

2.5.4 Robotics

Wooldridge and Jennings do not consider robotics as an origin [197]. However, the description of a robot, as given by Chapek [39] (the author that coined the term "robot"), fulfils all four characteristics of an agent. Robots can be seen as agents and research in robotic architectures is a significant contributor to the agent paradigm.

The aim of robotics is to develop a machine that can assist humans. Robotics-related research of agent systems can be divided into two main groups, namely simulated robot systems and physical robots. These two groups are described next.

Simulated Robot Systems

Research in simulated robot systems is closely related to the field of AI and Distributed Artificial Intelligence (DAI). Distributed Problem Solving (DPS) is a sub-field of DAI, and in a way, multiple cooperating robots systems can be seen as a special case of distributed systems [38]. Simulated robot systems can be divided into two classes [113]: those that simulate situated agents and those that simulate abstract agents.

Simulated situated agents are embedded in simulated environments. One of the roles of simulated situated agents is that of a very valuable tool for making decisions on the design of physical robots. If the simulation environment accurately caters for physical laws and constraints, then design decisions can be made based on the results of simulations. Examples of design decisions are choice of sensors, sensor positioning, means of locomotion, etc. Other roles of simulated situated agents include accurate overall evaluation of a proposed physical robot system and experimentation on large-scale systems, which include a larger number of agents. A good example of a simulated robot system is given in [85].

Simulated robot systems that simulate abstract agents are useful for experimenting with aspects of robotic systems that are not related to robots' interaction with the environment, and as such have a limited role. Simulated robot systems that simulate abstract agents usually use a very high level of abstraction when interacting with the simulated environment. For example, a simulated robot system would assume that tasks such as "recognise-object" are atomic. From a DAI point of view, a simulated robot system with a high level of abstraction can be used to test cooperation and communications models, and in more general terms any biological and sociological aspect of MAS.

- *Physical Robot Systems*

Building physical robot systems as a MAS research vehicle is a substantial engineering task that, until recently, was attempted by a relatively small number of researchers. For the purpose of this thesis an overview, by no means exhaustive, of some of the seminal physical robot systems is presented.

Arguably, the earliest agent-related physical robot was Shakey, developed at Stanford Research Institute [153][136]. Shakey was equipped with the Stanford Research Institute Problem Solver (STRIPS) [65], a symbolic planner system. Various insights were gained; probably the most important is that mapping of a real-world environment to a symbolic world model is far from trivial. It has been observed that not all algorithms that perform well in simulated environments succeed in embodied systems. More details on Shakey can be found in [120].

In the '90s numerous robots based on the work of Rodney Brooks [31] were designed and implemented, for example, Myrmix [44]. Myrmix is a simple robot that has only three layers; each of the layers representing a simple action such as "collect", "avoid obstacle" and "safe forward". Genghis, also based on the work of Brooks [30], demonstrates how a simple architecture (the subsumption architecture) can achieve a relatively complex task, namely walking on six legs.

Behaviour based robotics, which developed from the subsumption architecture, addressed some of the shortcomings of the subsumption architecture. The shortcomings that were addressed include the lack of learning mechanism and lack of communicating mechanisms. Matarić, one of the foremost researchers in this field, has developed numerous physical robot systems based on the behaviour based robotics paradigm [113].

Another novel approach is that of the robotic ecosystem developed by McFarland and Steels [121], where the idea was to observe and facilitate emergence of cooperation between robots. Others, such as Aylett *et al* [11][10], created a hybrid architecture where a behaviour based architecture was implemented in robots and a symbolic planner component was implemented in a desktop computer. Arguably, the most important impetus to renewed interest in robotics research stemmed from the establishment of RoboCup [96]. Pfeiefer *et al* [148] have argued that the impetus that RoboCup has given to robotics can be compared to the impetus that the Apollo program gave to the exploration of space.

The renewed interest resulted in a large number of robot systems that have appeared over the last decade. Furthermore, the public interest in robotics has increased and as a result, numerous robotic kits are available today [185][184].

This thesis assumes learning and cooperation to be the key characteristics of an agent. One fairly recent physical robot system that emphasises these two characteristics deserves mention here, namely ALLIANCE [144] and its evolution, L-ALLIANCE [145]. Although its results do not exceed hand-crafted solutions, the system exhibited learning behaviour [142].

2.6 Summary

This chapter overviewed and discussed the various agent and MAS definitions as well as the origins of the agent and MAS paradigm. Some of the problems related to the MAS paradigm were also discussed. Various classification schemes were overviewed and the reasoning model classification scheme was adopted for the purpose of this thesis.

The next chapter provides a more detailed overview of agent architectures, classified according to the reasoning model used.